

## PATENT ABSTRACTS OF JAPAN

(11)Publication number : 10-307840

(43)Date of publication of application : 17.11.1998

BEST AVAILABLE COPY

(51)Int.Cl. G06F 17/30

(21)Application number : 09-119869

(71)Applicant : CANON INC

(22)Date of filing : 09.05.1997

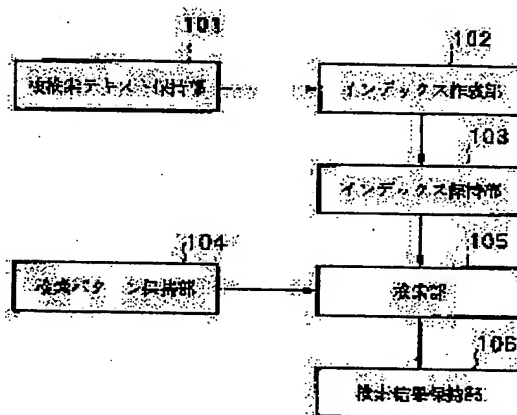
(72)Inventor : ITO SHIRO  
IKEDA YUJI

## (54) INFORMATION PROCESSOR AND ITS METHOD

## (57)Abstract:

PROBLEM TO BE SOLVED: To provide an information processor and its method capable of suppressing the increment of index keys to be used for the retrieval of text data and improving retrieving speed.

SOLUTION: A retrieved text storing part 101 stores text data and an index preparing part 102 prepares indexes related to the positions of character strings constituting text data based on character strings appearing more than the prescribed number of times out of character strings constituting the stored text data. A retrieving part 105 retrieves text data having an inputted retrieving pattern by using these prepared indexes and outputs the retrieved result to a retrieved result storing part 106.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平10-307840

(43) 公開日 平成10年(1998)11月17日

(51) Int.Cl.<sup>6</sup>

G 0 6 F 17/30

識別記号

F I

G 0 6 F 15/413

15/40

15/401

3 1 0 B

3 7 0 A

3 1 0 A

審査請求 未請求 請求項の数16 O L (全 15 頁)

(21) 出願番号

特願平9-119869

(22) 出願日

平成9年(1997)5月9日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 伊藤 史朗

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72) 発明者 池田 裕治

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

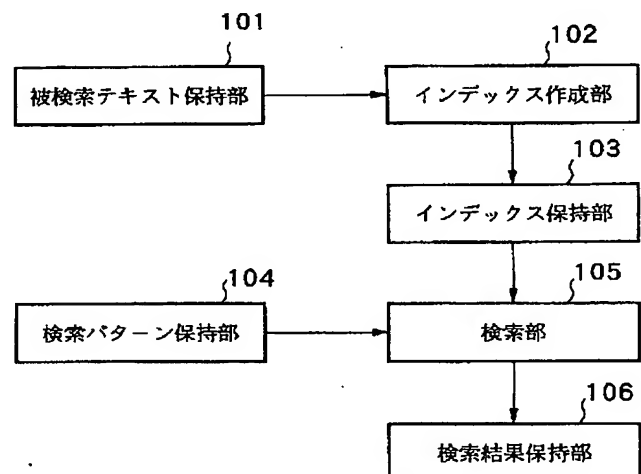
(74) 代理人 弁理士 大塚 康徳 (外1名)

(54) 【発明の名称】 情報処理装置及びその方法

(57) 【要約】

【課題】 テキストデータの検索に用いるインデックスのキーの増大を抑えるとともに、検索速度を向上することができる情報処理装置及びその方法を提供する。

【解決手段】 テキストデータを被検索テキスト保持部101に保持し、保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関するインデックスをインデックス作成部102で作成する。作成されたインデックスを用いて、入力された検索パターンを有するテキストデータを検索部105で検索する。そして、検索結果を検索結果保持部106に出力する。



## 【特許請求の範囲】

【請求項 1】 テキストデータを検索する情報処理装置であって、  
 テキストデータを保持する保持手段と、  
 前記保持手段で保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成手段と、  
 前記作成手段で作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する検索手段と、  
 前記検索手段による検索結果を出力する出力手段とを備えることを特徴とする情報処理装置。

【請求項 2】 前記作成手段は、前記所定回数以上出現する文字列と所定の位置関係を有する文字列の位置に関する位置情報を作成することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 3】 前記作成手段は、前記所定回数以上出現する文字列に前接する文字列の位置に関する位置情報を作成することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 4】 前記作成手段は、前記所定回数以上出現する文字列に後接する文字列の位置に関する位置情報を作成することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 5】 前記テキストデータを構成する文字列は、1 文字を含むことを特徴とする請求項 1 に記載の情報処理装置。

【請求項 6】 前記作成手段は、前記テキストデータ中で、前記所定回数以上出現する文字列以外の文字列の位置に関する位置情報を作成し、該位置情報を一括して管理することを特徴とする請求項 1 に記載の情報処理装置。

【請求項 7】 テキストデータを管理する情報処理装置であって、  
 入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成手段と、  
 前記作成手段で作成された位置情報と前記テキストデータを対応づけて管理する管理手段とを備えることを特徴とする情報処理装置。

【請求項 8】 テキストデータを検索する情報処理方法であって、  
 テキストデータを記憶媒体に保持する保持工程と、  
 前記保持工程で前記記憶媒体に保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程と、  
 前記作成工程で作成された位置情報を用いて、入力され

2

た検索パターンを有するテキストデータを検索する検索工程と、  
 前記検索工程による検索結果を出力する出力工程とを備えることを特徴とする情報処理方法。

【請求項 9】 前記作成工程は、前記所定回数以上出現する文字列と所定の位置関係を有する文字列の位置に関する位置情報を作成することを特徴とする請求項 8 に記載の情報処理方法。

【請求項 10】 前記作成工程は、前記所定回数以上出現する文字列に前接する文字列の位置に関する位置情報を作成することを特徴とする請求項 8 に記載の情報処理方法。

【請求項 11】 前記作成工程は、前記所定回数以上出現する文字列に後接する文字列の位置に関する位置情報を作成することを特徴とする請求項 8 に記載の情報処理方法。

【請求項 12】 前記テキストデータを構成する文字列は、1 文字を含むことを特徴とする請求項 8 に記載の情報処理方法。

【請求項 13】 前記作成工程は、前記テキストデータ中で、前記所定回数以上出現する文字列以外の文字列の位置に関する位置情報を作成し、該位置情報を一括して管理することを特徴とする請求項 8 に記載の情報処理方法。

【請求項 14】 テキストデータを管理する情報処理方法であって、  
 入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程と、  
 前記作成工程で作成された位置情報と前記テキストデータを対応づけて記憶媒体に管理する管理工程とを備えることを特徴とする情報処理方法。

【請求項 15】 テキストデータを検索する情報処理のプログラムコードが格納されたコンピュータ可読メモリであって、  
 テキストデータを記憶媒体に保持する保持工程のプログラムコードと、  
 前記保持工程で前記記憶媒体に保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程のプログラムコードと、  
 前記作成工程で作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する検索工程のプログラムコードと、  
 前記検索工程による検索結果を出力する出力工程のプログラムコードとを備えることを特徴とするコンピュータ可読メモリ。

【請求項 16】 テキストデータを管理する情報処理の

プログラムコードが格納されたコンピュータ可読メモリであって、  
 入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程のプログラムコードと、  
 前記作成工程で作成された位置情報と前記テキストデータを対応づけて記憶媒体に管理する管理工程のプログラムコードとを備えることを特徴とするコンピュータ可読メモリ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、テキストデータを検索する情報処理装置及びその方法に関するものである。

【0002】

【従来の技術】文書中の全てのテキストデータを対象として与えられた検索パターンを含むテキストデータを検索する全文検索装置などの情報処理装置では、大量のテキストデータを高速に検索するために、被検索テキストデータのインデックスを予め作成し、作成したインデックスを用いて検索を行なうことが多い。こうしたインデックス方式の一つに、*n*-gramインデックス方式と呼ばれる検索方式がある。*n*-gramインデックス方式とは、テキストデータ中の連続する*n*文字をキーとして、キーとなる文字列の被検索テキストデータ中での存在位置を列挙した位置リストを保持するインデックスである。

【0003】例えば、*n*を2に固定すると、文字セット*C*に対して、文字列集合 $S = \{c \mid c \in C\} \cup \{c_1c_2 \mid c_1, c_2 \in C\}$ を定義する。文字セット*C*から構成される検索対象文書*T*において、*T*における文字列 $s \in S$ の出現回数を $ns$ とし、*T*において $s$ が*x*回目に出現するときの $s$ を構成する先頭文字の位置を $pos(s, x)$ で表すとする。例えば、2-gramインデックスのキー*k*は、 $ns \geq 1$ である*s*になる。そして、*k*に対する位置リスト*Lk*は、

$Lk = (pos(k, 1), pos(k, 2), \dots, pos(k, nk))$

となる。2-gramインデックスは、(*k*, *Lk*)を全ての*k*について集めたものである。例えば、「日本対スイスの日本側当日券は本日売り切れ」のテキストデータに対する2-gramインデックスは図8のようになる。

【0004】次に、*N*-gramインデックスを用いた検索処理について説明する。ここでも、*n*を2に固定した場合で説明するが、*n*の値が変わっても基本は同じである。被検索テキストデータ*T*から文字長*l*の検索パターン*q*を検索する処理は、以下になる。ここで、 $s_i$ は、検索パターンの*i*番目から始まる2文字の部分

文字列を示す。但し、*s*<sub>1</sub>の長さだけは1とする。

【0005】1. インデックス*I*において、検索パターン*q*中の部分文字列 $s_i$  ( $i = 2j + 1$ , *l*が奇数のとき  $0 \leq j \leq (l-1)/2$ , *l*が偶数のとき  $0 \leq j \leq l/2 - 1$ ) に関して、 $ns_i$ の小さい順に応じて*i*を並べた数列( $t(0), t(1), \dots, t(m)$ )を作成する。ここで、 $m = (l-1)/2$  (*l*が奇数の場合)、 $m = l/2 - 1$  (*l*が偶数の場合)になる。

【0006】2.  $nst(0) = 0$ ならば検索処理は終了する。検索パターンは存在しない。

3.  $kj = st(j)$  ( $0 \leq j \leq m$ ) とする。

4.  $R(0) = \{(p - t(0) + 1) \mid p \in Lk0\}$  とする。

5.  $j = 1, 2, \dots, m$ まで、次の処理を繰り返す。途中で*R*(*j*)が空になったら検索処理は終了する。検索パターンは存在しない。

【0007】 $R(j) = \{(p - t(j) + 1) \mid p \in Lkj, \exists r \in R(j-1), r = p - t(j) + 1\}$

6. 検索処理は終了する。*R*(*m*)が検索パターンの存在する位置である。例えば「当日券」の検索処理は以下のようにになる。

1.  $s_1 = \text{当日}$ ,  $s_3 = \text{券}$ であるから、 $ns_1 = 1$ ,  $ns_3 = 1$ であり、 $t(0) = 1$ ,  $t(1) = 3$ となる。

【0008】2.  $ns_1 = 1$ で、0ではないので処理を続ける。

3.  $k_0 = \text{当日}$ ,  $k_1 = \text{券}$ とする。

4.  $Lk_0 = (11)$ であるから、 $R(0) = (11)$ となる。

5.  $Lk_1 = (13)$ であるから、 $l_1 = (13 - 3 + 1)$ を満足し、 $R(1) = (11)$ とする。

【0009】6. これにより、(11)が検索結果となる。  
*N*-gramインデックスの*n*の値を大きくすると、上記検索処理手順の第5ステップでの比較演算回数が減少するので、検索にかかる時間が短縮される。一方、*n*を大きくすると、インデックスサイズが増大したり、インデックスの作成時間が増大するという問題点がある。比較演算回数が増える要因の一つに、検索に使用するキーの出現回数(そのキーの位置リストの要素数)が大きいことがある。そこで、インデックスサイズを増大を抑えながら、検索処理時間の短縮を図るために、キーの出現回数が大きくなるようなキーについてのみ*n*の値を大きくするという方法がある。

【0010】こうした方法の一つとして、菅谷他：「*n*-gram型大規模全文検索方式の開発—インクリメンタル型*n*-gramインデックス方式—」(第53回情報処理学会全国大会論文集, 3, pp. 235-236)で説明されているインクリメンタル法がある。このインクリメンタル法では、*n*-gramのキーの出現回数が閾値を越えたら、そのキーの最後に1文字追加した(*n* + 1)-gramを作成して、それらをキーに追加する。例えば、キーの集合を*K*、閾値を*ta*としたとき、キー $k \in K$  (出現回数を $nk$ とする)に対して $nk > ta$

となったら、 $K' = K \cup \{s \mid s = \text{cat}(k, c), c \in K\}$  を新しいキー候補集合とする。ここで、 $\text{cat}(s, c)$  は、文字列  $s$  の後ろに文字  $c$  を加えた文字列を示す。このインクリメンタル法を用いて、「日本対スイスの日本側当日券は本日売り切れ」のテキストデータに対して閾値を1としてインデックスを作成すると図9のようになる。

【0011】別の方法として、福島他「高速全文検索のためのフレキシブル文字列インバージョン法(1)方式概要」(第53回情報処理学会全国大会論文集, 3, p. 239-240)で説明されているフレキシブル文字列インバージョン法がある。このフレキシブル文字列インバージョン法では、 $n\text{-gram}$ のキーの出現回数が閾値を越えた場合に、 $(n+1)\text{-gram}$ を作成する代わりに、キーの後(前)に接続する文字にハッシングを施し、得られたハッシュ値と元のキーを組み合わせたものを新たなキーとする。例えば、キーの集合を  $K$ 、閾値を  $t_b$  としたとき、キー  $k \in K$  に対して  $n k > t_b$  となったら、 $K' = K \cup \{h \mid h = \text{key}(k, \text{hash}(c)), c \in K\}$  を新しいキー候補集合とする。ここで、 $\text{hash}(c)$  は、文字  $c$  にハッシュを施したハッシュ値を示す。また、 $\text{key}(s, i)$  は、文字列  $s$  と整数値  $i$  とで構成されるキーを示す。このフレキシブル文字列インバージョン法を用いて、「日本対スイスの日本側当日券は本日売り切れ」のテキストデータに対して閾値を1としてインデックスを作成すると図10のようになる。尚、図中#で示している数値がハッシュ値である。

#### 【0012】

【発明が解決しようとする課題】しかしながら、上記従来のインクリメンタル法を用いた情報処理装置では、閾値を越えたキーに対して任意の文字を付け加えて新しいキーを作成するためキーの数が増えすぎ、ひいてはインデックスサイズを増大させるという問題があった。

【0013】また、上記従来のフレキシブル文字列インバージョン法を用いた情報処理装置では、ハッシュ値と組み合わせたキーを用いることで検索に使用するキーの位置リスト要素数は削減されるものの、組み合わせ演算の回数を削減できず、ひいては検索時間が十分に短縮されないという問題があった。この問題は、次の例をみるとわかりやすい。

【0014】今、アルファベットだけを対象とした場合に、キー「あ」の出現回数が閾値を越えたとする。この場合、インクリメンタル法では、新たにキー「ああ」、「あい」、「あう」、…、「あん」が作成される。このように、インクリメンタル法では、新しく作成されるキーの数が大きくなる。一方、フレキシブル文字列インバージョン法では、ハッシュ値が0と1になるようにハッシングを施したとすると、キー「あ#0」、「あ#1」が新しく作成されるだけで増大するキーの数を小さく抑

えられる。次に、「あいう」という検索パターンで検索することを考える。インクリメンタル法では、検索に使用するキーが「あい」と「う」となり、この二つの組み合わせ演算だけで検索が可能である。一方、フレキシブル文字列インバージョン法では、「あ#0」、「い」、「う」の三つの組み合わせ演算が必要になる(ここでは、「い」のハッシュ値が0になると仮定している)。なぜなら、キー「あ#0」だけでは、「あ」の次の文字が「い」である保障はないからである。

【0015】本発明は上記の問題に鑑みてなされたものであり、テキストデータの検索に用いるインデックスのキーの増大を抑えるとともに、検索速度を向上することができ、情報処理装置及びその方法を提供することを目的とする。

#### 【0016】

【課題を解決するための手段】上記の目的を達成するための本発明による情報処理装置は以下の構成を備える。即ち、テキストデータを検索する情報処理装置であって、テキストデータを保持する保持手段と、前記保持手段で保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成手段と、前記作成手段で作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する検索手段と、前記検索手段による検索結果を出力する出力手段とを備える。

【0017】また、好ましくは、前記作成手段は、前記所定回数以上出現する文字列と所定の位置関係を有する文字列の位置に関する位置情報を作成する。また、好ましくは、前記作成手段は、前記所定回数以上出現する文字列に前接する文字列の位置に関する位置情報を作成する。また、好ましくは、前記作成手段は、前記所定回数以上出現する文字列に後接する文字列の位置に関する位置情報を作成する。

【0018】また、好ましくは、前記テキストデータを構成する文字列は、1文字を含む。また、好ましくは、前記作成手段は、前記テキストデータ中で、前記所定回数以上出現する文字列以外の文字列の位置に関する位置情報を作成し、該位置情報を一括して管理する。上記の目的を達成するための本発明による情報処理装置は以下の構成を備える。即ち、テキストデータを管理する情報処理装置であって、入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成手段と、前記作成手段で作成された位置情報と前記テキストデータを対応づけて管理する管理手段とを備える。

【0019】上記の目的を達成するための本発明による情報処理方法は以下の構成を備える。即ち、テキストデータを検索する情報処理方法であって、テキストデータ

を記憶媒体に保持する保持工程と、前記保持工程で前記記憶媒体に保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程と、前記作成工程で作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する検索工程と、前記検索工程による検索結果を出力する出力工程とを備える。

【0020】上記の目的を達成するための本発明による情報処理方法は以下の構成を備える。即ち、テキストデータを管理する情報処理方法であって、入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程と、前記作成工程で作成された位置情報と前記テキストデータを対応づけて記憶媒体に管理する管理工程とを備える。

【0021】上記の目的を達成するための本発明によるコンピュータ可読メモリは以下の構成を備える。即ち、テキストデータを検索する情報処理のプログラムコードが格納されたコンピュータ可読メモリであって、テキストデータを記憶媒体に保持する保持工程のプログラムコードと、前記保持工程で前記記憶媒体に保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程のプログラムコードと、前記作成工程で作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する検索工程のプログラムコードと、前記検索工程による検索結果を出力する出力工程のプログラムコードとを備える。

【0022】上記の目的を達成するための本発明によるコンピュータ可読メモリは以下の構成を備える。即ち、テキストデータを管理する情報処理のプログラムコードが格納されたコンピュータ可読メモリであって、入力されたテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する作成工程のプログラムコードと、前記作成工程で作成された位置情報と前記テキストデータを対応づけて記憶媒体に管理する管理工程のプログラムコードとを備える。

【0023】

【発明の実施の形態】以下、図面を参照して本発明の好適な実施形態を詳細に説明する。図1は本発明の実施形態に係る情報処理装置の機能構成を示すブロック図である。図1において、101は被検索テキスト保持部であり、被検索テキストデータを保持する。102はインデックス作成部であり、被検索テキスト保持部101に保持されている被検索テキストデータに対して、キー（テキストデータを構成する文字あるいは文字列）集合に属するキーに対して被検索テキストデータ中での当該キー

の出現位置を列挙したインデックスを作成する。また、出現回数が基準以上のキーに1文字を付与したキー候補の中、被検索テキストデータ中での出現回数が別の基準以上のキー候補をキー集合に加えて、同様に被検索テキストデータ中での当該キーの出現位置を列挙したインデックスを作成する。103はインデックス保持部であり、インデックス作成部102で作成したインデックスを保持する。

【0024】104は検索パターン保持部であり、被検索テキストデータから検索するパターンを保持する。105は検索部であり、インデックス保持部103に保持されているインデックスを用いて、検索パターン保持部104に保持されている検索パターンを被検索テキストデータ中から検索する。106は検索結果保持部であり、検索部105による検索結果を保持する。

【0025】次に本発明の実施形態の情報処理装置の構成について、図2を用いて説明する。図2は本発明の実施形態の情報処理装置の構成を示すブロック図である。図2において、201はCPUであり、後述する手順を実現するプログラムに従って動作する。202はRAMであり、被検索テキスト保持部101、検索パターン保持部104、検索結果保持部106と上記プログラムの動作に必要な記憶領域とを提供する。203はROMであり、後述する手順を実現するプログラムを保持する。204はディスク装置であり、インデックス保持部103を実現する。205は情報処理装置の各種構成要素を相互に接続するバスである。206はキーボード及びマウスからなる入力装置であり、検索キーを入力する。207は例えば、CRT、LCD等の出力装置であり、検索結果を表示する。

【0026】以下、説明していく本発明の実施形態で実行される処理は、インデックスを作成するインデックス作成処理と文字列を検索する検索処理の2つに大きく分かれる。まず、インデックス作成処理によって作成されるインデックスを保持するインデックス保持部103の詳細な構成について、図11を用いて説明する。

【0027】図11は本発明の実施形態のインデックス保持部の詳細な構成を示す図である。図11において、1101は拡張コード領域1103に示される拡張コードに1文字を後接したキーが被検索テキストデータ中に出現する回数を保持する出現回数領域である。この出現回数領域では、行が拡張コードに対応し、列が後接する文字に対応する2次元の出現回数テーブルを保持する。但し、0行目については元になるキーがなく、各列が示す文字だけのキーの出現回数を示す。

【0028】1102は、拡張コードに1文字を後接したキーに対し、位置リストを保持する際のキーコードを保持するキーコード領域である。このキーコード領域では、行が拡張コードに対応し、列が後接する文字に対応する2次元のキーコードテーブルを保持する。但し、0

行目については元になるキーがなく、各列が示す文字だけのキーのキーコードを示す。また、0列目については元になるキーの位置リストから、文字を後接して作成した位置リストに含まれる要素を除いて作成する位置リストに対するキーコードを保持する。

【0029】1103は、拡張コードに1文字を後接したキーをさらに1文字拡張する場合の管理を行なう拡張コードを保持する拡張コード領域である。この拡張コード領域では、行が拡張コードに対応し、列が後接する文字に対応する2次元の拡張コードテーブルを保持する。但し、0行目については元になるキーがなく、各列が示す文字だけのキーの拡張コードを示す。

【0030】1104は、キーコードが示すキーに対する位置リストを保持する位置リスト領域である。キーコード値の昇順に並んでいるので、キーコード値から容易に位置リストへアクセスできる。以下の説明では、出現回数テーブルにおけるn行の文字cに対応する列の値をC(n, c)で表し、キーコードテーブルにおけるn行の文字cに対応する列の値をK(n, c)で表し、キーコードテーブルにおけるn行の0列目の値をK0(n)で表し、拡張コードテーブルにおけるn行の文字cに対応する列の値をE(n, c)で表し、キーコードKに対する位置リストをL(k)で表す場合もある。

【0031】次に、インデックス作成部102で実行されるインデックス作成処理について、図3を用いて説明する。図3は本発明の実施形態で実行されるインデックス作成処理を示すフローチャートである。まず、ステップS301では、出現回数テーブル、キーコードテーブル、拡張コードテーブルの全要素を全て0に初期化する。また、処理の対象となっている文字の位置を示すカウンタCを1に初期化する。次に、ステップS302では、ポインタPの初期化を行なう。ポインタPは、処理の対象となっている文字を指し示すもので、これを被検索テキストデータの先頭文字を指すように初期化する。ステップS303では、ポインタpが被検索テキストデータの最後に達したか否かを判定する。最後に達している場合(ステップS303でYES)、インデックス作成処理を終了する。一方、最後に達していない場合(ステップS303でNO)、ステップS304に進む。

【0032】ステップS304では、引数(0, p, 1)による登録処理を行なう。尚、この登録処理の詳細については、図4を用いて後述する。また、(0, p, 1)は図4のフローチャートを呼び出すための引数である。ステップS305では、カウンタcの値を1増やす。ステップS306では、ポインタpが次の文字を指し示すようポインタpを進め、ステップS303に戻る。

【0033】次に、上述した図3のステップS304における引数(n, p, s)への登録処理の詳細について、図4を用いて説明する。図4は本発明の実施形態で

実行される登録処理の詳細を示すフローチャートである。まず、ステップS401では、ポインタpが指し示す文字をcとする。ステップS402では、出現回数テーブルにおいて、n行の文字cに対応する列C(n, c)の値を1増やす。ステップS403では、引数(n, c, s)に対する新キー作成処理を行なう。尚、この新キー作成処理の詳細については、図5を用いて後述する。ステップS404では、キーコードテーブルにおいて、n行の文字cに対応する列K(n, c)の値を参照する。列K(n, c)の値が0である場合は、ステップS405に進み、キーコードテーブルにおけるn行の0列目の値をキーコードとして、当該キーコードに対する位置リストL(K0(n))にカウンタCの値を追加する。そして、ステップS407に進む。一方、列K(n, c)の値が正の値である場合は、ステップS406に進み、キーコードテーブルにおいて、n行の文字cに対応する列の値をキーコードとして、当該キーコードに対する位置リストL(K0(n), c)にカウンタCの値を追加する。そして、ステップS407に進む。

【0034】ステップS407では、拡張コードテーブルにおいて、n行の文字cに対応する列E(n, c)の値を参照する。列E(n, c)の値が0である場合は、(n, p, s)への登録処理を終了する。一方、列E(n, c)の値が正の値である場合は、ステップS408に進む。ステップS408では、ポインタpが次の文字を指し示すようポインタpを進める。ステップS409では、ポインタpが検索テキストデータの最後に達したか否かを判定する。最後に達した場合(ステップS409でYES)、(n, p, s)への登録処理を終了する。一方、最後に達していない場合(ステップS409でNO)、ステップS410に進む。

【0035】ステップS410では、拡張コードテーブルにおいて、n行の文字cに対応する列E(n, c)の値をmとして、引数(m, p, (s+1))への登録処理を再帰し、引数(m, p, (s+1))への登録処理を終了する。次に、上述した図4のステップS403における引数(n, c, s)に対する新キー作成処理の詳細について、図5を用いて説明する。

【0036】図5は本発明の実施形態で実行される新キー作成処理の詳細を示すフローチャートである。まず、ステップS501では、出現回数テーブルにおいて、n行の文字cに対応する列C(n, c)の値を参照し、C(n, c)の値が閾値Aを越えており、かつ拡張コードテーブルにおいて、n行の文字cに対応する列E(n, c)の値を参照し、列E(n, c)の値が0あるという条件を満たすか否かを判定する。条件を満たす場合(ステップS501でYES)、ステップS508に進む。一方、条件を満たさない場合(ステップS501でNO)、ステップS502に進む。

【0037】ステップS502では、nの値を参照し、



値が0であるか否かを判定する。値が0である場合（ステップS502でYES）、ステップS503に進む。一方、値が0でない場合（ステップS502でNO）、ステップS505に進む。ステップS503では、キーコードテーブルにおいて、n行の文字cに対応する列K(n, c)の値が0であるか否かを判定する。列K(n, c)の値が0である場合（ステップS503でYES）、ステップS504に進み、新しいキーコードを割り当て、割り当てた値をキーコードテーブルにおけるn行の文字cに対応する列K(n, c)に代入する。そして、引数(n, c, s)に対する新キー作成処理を終了する。一方、列K(n, c)の値が0でない場合（ステップS503でNO）、(n, c, s)に対する新キー作成処理を終了する。

【0038】ステップS505では、出現回数テーブルにおいて、n行の文字cに対応する列C(n, c)の値を参照し、列C(n, c)値が閾値Bを越えており、かつキーコードテーブルにおいて、n行の文字cに対応する列K(n, c)の値を参照し、列K(n, c)値が0であるという条件を満たすか否かを判定する。条件を満たす場合（ステップS505でYES）、ステップS506に進む。一方、条件を満たさない場合（ステップS505でNO）、引数(n, c, s)に対する新キー作成処理を終了する。

【0039】ステップS506では、新しいキーコードを割り当て、割り当てた値をキーコードテーブルにおけるn行の文字cに対応する列K(n, c)に代入する。ステップS507では、キーコードテーブルにおけるn行の0列の値をキーとする位置リスト中の各値について、その値にsを足した値の中から、キーコードテーブルにおけるn行の文字cに対応する列の値をキーとする位置リスト中の値のいずれかに一致する値を全て削除する。次に、削除した値をステップS506で新たに割り当てたキーコードに対する位置リストに加える。そして、引数(n, c, s)に対する新キー作成処理を終了する。

【0040】ステップS508では、新しい拡張コードを割り当て、割り当てた値mを拡張コードテーブルにおけるn行の文字cに対応する列E(n, c)に代入する。ステップS509では、新しいキーコードを割り当て、割り当てた値をキーコードテーブルにおけるm行の0列に代入する。そして、キーコードテーブルにおけるn行の文字cに対応する列の値をキーとする位置リストの内容を、全て割り当てたキーコードの位置リストにコピーする。

【0041】ステップS510では、キーコードテーブルの全ての列において、以下の処理を行なう。m行の0列の値をキーとする位置リスト中の各値について、その値にsを足した値の中からキーコードテーブルにおけるn行の当該列の値をキーとする位置リスト中の値のい

れかに一致する値の数を調べ、それが閾値Bを越えていれば一致した値を全て削除する。次に、新しいキーコードを割り当て、削除した値を割り当てたキーコードの位置リストに全て加え、キーコードテーブルにおけるm行の当該列の値に割り当てたキーコードを代入する。そして、引数(n, c, s)に対する新キー作成処理を終了する。

【0042】以上のインデックス作成処理を、例えば、「日本対スイスの日本側当日券は本日売り切れ」に対して施すと、図11に示すようなインデックスが作成される。また、図12は、「当日券」の「日」までのインデックス作成処理が完了した時点でのインデックスを示している。次に検索部105で実行される検索処理について、図6を用いて説明する。

【0043】図6は本発明の実施形態で実行される検索処理を示すフローチャートである。まず、ステップS601では、検索パターン保持部104に保持されている検索パターンの長さを演算用領域1に代入する。また、演算用領域nに1を代入する。ステップS602では、演算用領域nに対する位置リスト獲得処理を行ない、獲得した位置リストを演算用配列A1に格納する。尚、この位置リスト獲得処理の詳細については、図7を用いて説明する。

【0044】ステップS603では、演算用領域nの示す値が演算用領域1の示す値より大きいかなんかを判定する。演算用領域nの示す値が演算用領域1の示す値より大きい場合（ステップS603でYES）、ステップS606に進む。一方、演算用領域nの示す値が演算用領域1の示す値未満である場合（ステップS603でNO）、ステップS604に進む。

【0045】ステップS604では、演算用領域nに対する位置リスト獲得処理を行ない、獲得した位置リストを演算用配列A2に格納する。ステップS605では、演算用配列A1と演算用配列2の両方に存在している値を全て取り出し、これらの値だけからなる位置リストを新たに演算用配列A1に格納する。そして、ステップS603に戻る。

【0046】ステップS606では、演算用配列A1が空であるか否かを調べる。空でない場合（ステップS606でNO）、ステップS607に進み、被検索テキストデータから検索パターンが検索されたことを示す値として1を検索結果保持部106に保持する。そして検索処理を終了する。一方、空である場合（ステップS606でYES）、ステップS608に進み、被検索テキストデータから検索パターンが検索されなかったことを示す値として0を検索結果保持部106に保持する。そして、検索処理を終了する。

【0047】次に、上述した図6のステップS602における位置リスト獲得処理の詳細について、図7を用いて説明する。図7は本発明の実施形態で実行される位置



リスト獲得処理の詳細を示すフローチャートである。まず、ステップS701では、演算用領域mの値を演算用領域nの値に初期化する。また、演算用領域kの値を0に初期化する。ステップS702では、演算用領域mの値が演算用領域kの値以下であるか否かを判定する。演算用領域mの値が演算用領域kの値未満である場合（ステップS702でYES）、ステップ703に進む。一方、演算用領域mの値が演算用領域kの値より大きい場合（ステップS702でNO）、ステップS706に進む。

【0048】ステップS703では、拡張コードテーブルのk行で、検索パターンのm番目の文字に対応する列の値をk'に代入する。ステップS704では、k'の値が0であるか否かを判定する。k'の値が0である場合（ステップS704でYES）、ステップS706に進む。一方、k'の値が正の値である場合（ステップS704でNO）、ステップS705に進む。ステップS705では、演算用領域kにk'の値を代入し、演算用領域mの値を1増やす。そして、ステップS702に戻る。

【0049】ステップS706では、出現回数テーブルのk行で、検索パターンのm番目の文字に対応する列C(k, c)の値が0であるか否かを判定する。列C(k, c)の値が0である場合（ステップS706でYES）、ステップS707に進み、空の位置リストを獲得した位置リストとする。そして、ステップS711に進む。一方、列C(k, c)の値が0でない場合（ステップS706でYES）、ステップS708に進む。

【0050】ステップS708では、キーコードテーブルのk行で、検索パターンのm番目の文字に対応する列K(k, c)の値が0であるか否かを判定する。列K(k, c)の値が0である場合（ステップS708でYES）、ステップS709に進み、キーコードテーブルのk行で0列の値をキーとして、そのキーコードに対応する位置リストを取り出す。そして、その全ての要素から(n-1)を引いた位置リストを獲得した位置リストとする。続いて、ステップS712では、検索処理全般で使用する演算用領域nにmを代入する。そして、位置リスト獲得処理を終了する。

【0051】一方、列K(k, c)の値が0でない場合（ステップS708でNO）、ステップS710に進み、キーコードテーブルのk行で、検索パターンのm番目の文字に対応する列の値をキーとして、そのキーコードに対応する位置リストを取り出す。そして、その全ての要素から(n-1)を引いた位置リストを獲得した位置リストとする。続いて、ステップS711では、検索処理全般で使用する演算用領域nに(m+1)を代入する。そして、位置リスト獲得処理を終了する。

【0052】次に、検索パターン「当日券」で検索する場合の検索処理の具体例について説明していく。

1.  $l=3$ 、 $n=1$ を代入する。（ステップS601）
2.  $n=1$ で位置リスト獲得処理を行なう。（ステップS602）

(a)  $m=n=1$ 、 $k=0$ を代入する。（ステップS701）

(b)  $m < l$ なので（ステップS702でYES）、 $k'=E(0, \text{当})=0$ とする。（ステップS703）

(c)  $k'=0$ （ステップS704でYES）、 $C(0, \text{当})=1$ （ステップS706でNO）、 $K(0,$

10 当)=8（ステップS708でNO）

なので、 $L(8)=(11)$ から0を引いた(11)を獲得する位置リストとする。（ステップS710）

(d)  $n=m+1=2$ とする。（ステップS711）

3.  $A1=(11)$ とする。（ステップS602）

4.  $n \leq l$ なので（ステップS603でNO）、位置リスト獲得処理を行なう。（ステップS604）

(a)  $m=n=2$ 、 $k=0$ 。（ステップS701）

(b)  $m < l$ なので（ステップS702でYES）、 $k'=E(0, \text{日})=1$ とする。（ステップS703）

20 (c)  $k' > 0$ （ステップS704でNO）なので、 $k=1$ 、 $m=m+1=3$ とする。（ステップS705）

(d)  $m=1$ （ステップS702でNO）、 $C(1, \text{券})=1$ （ステップS706でNO）、 $K(1, \text{券})=0$ （ステップS708でYES）なので、 $L(K(1))=L(9)=(12, 16)$ から1を引いた(11, 15)を獲得する位置リストとする。（ステップS709）

(e)  $n=m=3$ とする。（ステップS712）

5.  $A2=(11, 15)$ とする。（ステップS604）

6.  $A1$ と $A2$ の両方に存在する要素をとり、 $A1=(11)$ とする。（ステップS605）

30

7.  $n \leq l$ なので（ステップS603でNO）、位置リスト獲得処理を行なう。（ステップS604）

(a)  $m=n=3$ 、 $k=0$ 。（ステップS701）

(b)  $m \geq l$ なので（ステップS702でNO）、 $C(0, \text{券})=1$ （ステップS706でNO）、 $K(0, \text{券})=11$ （ステップS708でNO）なので、 $L(11)=(13)$ から2を引いた(11)を獲得する位置リストとする。（ステップS710）

(c)  $n=m+1=4$ とする。（ステップS711）

8.  $A2=(11)$ とする。（ステップS604）

40

9.  $A1$ と $A2$ の両方に存在する要素をとり、 $A1=(11)$ とする。（ステップS605）

10.  $n > l$ なので（ステップS603でYES）、 $A1$ は空でない（ステップS606でNO）、検索パターンが検索される。（ステップS607）

次に別の検索パターン「日本人」で検索する場合の検索処理の具体例について説明していく。

【0053】

1.  $l=3$ 、 $n=1$ を代入する。（ステップS601）

50

2.  $n=1$ で位置リスト獲得処理を行なう。（ステップ

S602)

(a)  $m=n=1$ 、 $k=0$ を代入する。(ステップS701)

(b)  $m<1$ なので(ステップS702でYES)、 $k'=E(0, \text{日})=1$ とする。(ステップS703)

(c)  $k'>0$ (ステップS704でNO)、 $k=1$ 、 $m=m+1$ とする。

【0054】(ステップS705)

(d)  $m<1$ なので(ステップS702でYES)、 $k'=E(1, \text{本})=0$ とする。(ステップS703)

(e)  $k'=0$ (ステップS704でYES)、 $C(1, \text{本})=2$ (ステップS706でNO)、 $K(1, \text{本})=10$ (ステップS708でNO)なので、 $L(10)=(1,8)$ から0を引いた(1,8)を獲得する位置リストとする。(ステップS710)

(f)  $n=m+1=3$ とする。(ステップS711)

3.  $A1=(1,8)$ とする。(ステップS602)

4.  $n\leq 1$ なので(ステップS603でNO)、位置リスト獲得処理を行なう。(ステップS604)

(a)  $m=n=3$ 、 $k=0$ 。(ステップS701)

(b)  $m\geq 1$ なので(ステップS702でNO)、 $C(0, \text{人})=0$ (ステップS706でYES)、空リスト()を獲得する位置リストとする。(ステップS707)

(c)  $n=m+1=4$ とする。(ステップS711)

5.  $A2=()$ とする。(ステップS604)

6.  $A1$ と $A2$ の両方に存在する要素をとり、 $A1=()$ とする。(ステップS605)

7.  $n>1$ なので(ステップS603でYES)、 $A1$ は空であるので(ステップS606でYES)、検索パターンは検索されない。(ステップS608)

以上のように、本実施形態の情報処理装置の検索によれば、インクリメンタル法などを利用した従来の情報処理装置の検索に比べてキーの数を削減することができる。上述の具体例において、従来のインクリメンタル法では「日本」、「日券」、「日売」の3つのキーが作成されるのに対し、本実施形態では「日本」、「日(0)」の2つとなり、キーの数が削減される。

【0055】また、本実施形態では、作成するキーの数が削減されても、検索処理における組み合わせ演算数が大きく増えるわけではない。例えば、上述の「日本人」の検索例では、1つの閾値を用いた1回の組み合わせ演算で検索が完了している。これに対し、同様の閾値を使って、従来のフレキシブル文字列インバージョン法では、組み合わせ演算が2回必要になる。つまり、本実施形態では、従来に比べて、より高速に検索処理を実行することができる。

【0056】以上説明したように、本実施形態によれば、文字列としての一致だけではなく、インデックスのキー数の増大を抑えるとともに組み合わせ演算数も削減

することで、インデックスサイズを抑えながら、検索時間の短縮が可能になるという効果が得られる。尚、上記実施形態においては、キーを拡張する際に全ての文字を後接した文字列で候補を作成する場合について説明したが、これに限定されるものではない。例えば、文字単独での出現回数が閾値Cを越える文字だけを後接する候補としてもよい。これにより、インデックス作成処理の計算量が削減される。つまり、上述した実施形態の新キー作成処理において、出現回数テーブルの0行目の値が閾値C以下の列については、各ステップでの処理を行わなければよい。あるいは、各テーブルの列には、出現回数が閾値Cを越えた文字しか入れない方法でもよい。この場合、各テーブルの0行目は分離したデータ構造とする。尚、どちらの方法にしても、本実施形態では、文字の出現回数が閾値Cを越えた時点で、拡張するキーの再計算処理を更に加える必要がある。

【0057】また、上記実施形態においては、キーを拡張するにあたり新しいキーの出現回数が閾値Aを越えない場合は、残りをまとめたキーに一括して保持する場合について説明したが、これに限定されるものではない。新しいキーの出現回数が閾値Aを越えない場合は、保持しなくてもよい。すなわち、キーコードテーブルの0列目を作成しないことになる。また、拡張する前のキーを使用すれば検索は可能である。但し、組み合わせ演算を行なう際のリストの要素数が増えるので、検索時間が増大する。また、新キー作成処理では、ステップS507とステップS510で、新しく位置リストを作成する場合には、拡張する前の元のキーの位置リストから求めればよい。

【0058】また、上記実施形態においては、全ての文字について一括のインデックスを作成する場合について説明したが、これに限定されるものではない。ひらがな、カタカナ、漢字などの字種ごとにインデックスを分けてもよい。この場合、拡張する場合には同一字種の文字でのみ拡張してもよいし、他の字種の文字も含めて拡張してもよい。

【0059】また、上記実施形態においては、インデックスを作成しながらキーの拡張を行なう場合について説明したが、これに限定されるものではない。一度被検索テキストデータ中でのキーの出現回数を調べて、出現回数テーブルを作成する。そして、その作成された出現回数テーブルに基づいて、キーコードテーブル、拡張コードテーブルを作成してから、位置リストの作成を進めてもよい。

【0060】また、上記実施形態においては、拡張する前のキーのキー長を1とする場合について説明したが、これに限定されるものではない。全て2以上の一定値であってもよいし、キーの字種などによりキー長を変えてもよい。また、上記実施形態においては、拡張するキーのキー長の上限がない場合について説明したが、これに

限定されるものではない。拡張するキーのキー長に上限を設けてもよい。

【0061】また、上記実施形態においては、キーを拡張する際に、文字を後接する場合について説明したが、これに限定されるものではない。前接したり間隔をあけた位置の文字を定めるなど任意に拡張してよい。また、上記実施形態においては、2次元のテーブルを利用してインデックスを管理する場合について説明したが、これに限定されるものではない。例えば、図15に示すようなトライを用いても実現できる。

【0062】図15は本発明の他の実施形態のインデックスのデータ構造と作成例を示す図である。図15において、1501はルートノードである。ルートノード1501は2つのデータをノード1502、ノード1503にそれぞれ保持する。ノード1502は、そのノード1502をキーとした場合のキーの出現回数を示すデータである。ノード1503は、そのノード1503をキーとした場合のキーコードを示すデータである。トライでは、上位のノードから自ノードまで辿る間の文字を付加したものが自ノードのキーとなる。従って、例えば、ノード1504は、キー「日本」を表す。また、ノード1505は、「日」を拡張したキーで、実際に作成されたキーに含まれない位置リストを保持するキーを示す。尚、位置リストは、図11の位置リスト領域1104と同じになる。

【0063】また、上記実施形態においては、図11に示す各領域を予め十分な大きさと確保しておく場合について説明したが、これに限定されるものではない。必要に応じて領域を増大させてもよい。また、上記実施形態においては、閾値を固定した場合について説明したが、これに限定されるものではない。字種などにより異なる閾値を用いてもよい。また、被検索テキストデータの大きさなどに応じて閾値を変化させてもよい。

【0064】また、上記実施形態においては、出現回数テーブル、キーコードテーブル、拡張コードテーブルの各テーブルの列を求める際に文字コードを使用する場合について説明したが、これに限定されるものではない。文字に対して内部的なコードを割り当て、そのコードで列を参照してもよい。また、上記実施形態においては、出現回数テーブル、キーコードテーブル、拡張コードテーブルの各テーブルや位置リストを参照する際にコードを使用する場合について説明したが、これに限定されるものではない。各テーブルのアドレスを指し示すポインタなどを使用してもよい。

【0065】また、上記実施形態においては、被検索テキストデータ中に検索パターンが存在するか否かを検索結果として保持する場合について説明したが、これに限定されるものではない。被検索テキスト中の検索パターンの存在位置を検索結果として保持してもよい。この場合、図6で説明した検索処理のステップS606の配列

A1の値が検索パターンの存在位置を示すので、これを用いれば、検索パターンの存在位置を検索結果として保持することができる。

【0066】また、上記実施形態においては、単一の被検索テキストデータに対して検索を行なう場合について説明したが、これに限定されるものではない。複数のテキストデータに対し、検索パターンが含まれるテキストデータを検索するために用いてもよいことは言うまでもない。また、上記実施形態においては、インデックス作成処理と検索処理を同一の情報処理装置で行なう場合について説明したが、これに限定されるものではない。インデックス作成処理と検索処理を異なる情報処理装置で行ってもよい。この場合の各情報処理装置の機能構成について、図13と図14を用いて説明する。尚、図13に示す情報処理装置と、図14に示す情報処理装置は、例えば、ネットワーク回線で接続され互いにデータの授受を可能とする構成になっている。また、あるいはインデックス作成処理をCD-ROM等の記憶媒体に記憶しておき、その記憶媒体を図14に示す情報処理装置に搭載して互いにデータの授受を可能とする構成になっている。また、あるいは、検索処理をCD-ROM等の記憶媒体に記憶しておき、その記憶媒体を図13に示す情報処理装置に搭載して互いにデータの授受を可能とする構成になっている。

【0067】図13は本発明の他の実施形態に係る情報処理装置の機能構成を示すブロック図である。図13において、1301は被検索テキスト保持部であり、被検索テキストデータを保持する。1302はインデックス保持部であり、被検索テキスト保持部1301に保持されている被検索テキストデータに対して、キー集合に属するキーに対して被検索テキストデータ中での当該キーの出現位置を列挙すると共に、キーの出現回数が基準以上の場合に、当該キーに1文字を付与したキー候補の中から、被検索テキストデータ中での出現回数が別の基準以上の場合に、当該キーをキー集合に加えて、同様に被検索テキストデータ中での当該キーの出現位置を列挙したインデックスを作成する。1303はインデックス保持部であり、インデックス作成部1302で作成したインデックスを保持する。

【0068】図14は本発明の他の実施形態に係る情報処理装置の機能構成を示すブロック図である。図14において、1401はインデックス保持部であり、図13に示す情報処理装置で作成されたインデックスを保持する。1402は検索パターン保持部であり、被検索テキストデータから検索するパターンを保持する。1403は検索部であり、インデックス保持部1401に保持されているインデックスを用いて、検索パターン保持部1402に保持されている検索パターンを被検索テキストデータ中から検索する。1404は検索結果保持部であり、検索部1403による検索結果を保持する。

【0069】また、上記実施形態においては、被検索テキスト保持部101、検索パターン保持部104、検索結果保持部106をRAM202で、インデックス保持部102をディスク装置204で実現する場合について説明したが、これに限定されるものではなく、任意の記憶媒体を用いて実現してもよい。また、上記実施形態においては、各構成要素を同一の情報処理装置上で構成する場合について説明したが、これに限定されるものではなく、ネットワーク上に分散した情報処理装置などに分かれて各構成要素を構成してもよい。

【0070】また、上記実施形態においては、プログラムをROM203に保持する場合について説明したが、これに限定されるものではなく、任意の記憶媒体を用いて実現してもよい。また、同様の動作をする回路で実現してもよい。尚、本発明は、複数の機器（例えば、ホストコンピュータ、インタフェース機器、リーダ、プリンタ等）から構成されるシステムに適用しても、一つの機器からなる装置（例えば、複写機、ファクシミリ装置等）に適用してもよい。

【0071】また、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読出し実行することによっても、達成されることは言うまでもない。

【0072】この場合、記憶媒体から読出されたプログラムコード自体が上述した実施の形態の機能を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。プログラムコードを供給するための記憶媒体としては、例えば、フロッピディスク、ハードディスク、光ディスク、光磁気ディスク、CD-ROM、CD-R、磁気テープ、不揮発性のメモリカード、ROMなどを用いることができる。

【0073】また、コンピュータが読出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているOS（オペレーティングシステム）などが実際の処理の一部または全部を行い、その処理によって前述した実施の形態の機能が実現される場合も含まれることは言うまでもない。

【0074】更に、記憶媒体から読出されたプログラムコードが、コンピュータに挿入された機能拡張ボードやコンピュータに接続された機能拡張ユニットに備わるメモリに書き込まれた後、そのプログラムコードの指示に基づき、その機能拡張ボードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0075】本発明を上記記憶媒体に適用する場合、そ

の記憶媒体には、先に説明したフローチャートに対応するプログラムコードを格納することになるが、簡単に説明すると、図16、図17のメモリマップ例に示す各モジュールを記憶媒体に格納することになる。すなわち、図16に示す、少なくとも「保持モジュール」、「作成モジュール」、「検索モジュール」および「出力モジュール」の各モジュールのプログラムコードを記憶媒体に格納すればよい。

【0076】尚、「保持モジュール」は、テキストデータを記憶媒体に保持する。「作成モジュール」は、記憶媒体に保持されているテキストデータを構成する文字列の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する。「検索モジュール」は、作成された位置情報を用いて、入力された検索パターンを有するテキストデータを検索する。「出力モジュール」は、検索結果を出力する。

【0077】また、図17に示す、少なくとも「作成モジュール」および「管理モジュール」の各モジュールのプログラムコードを記憶媒体に格納すればよい。尚、「作成モジュール」は、入力されたテキストデータを構成する文字列の各文字の内、所定回数以上出現する文字列に基づいて、該テキストデータを構成する文字列の位置に関する位置情報を作成する。「管理モジュール」は、作成された位置情報とテキストデータを対応づけて記憶媒体に管理する。

【0078】

【発明の効果】以上説明したように、本発明によれば、テキストデータの検索に用いるインデックスのキーの増大を抑えるとともに、検索速度を向上することができる情報処理装置及びその方法を提供できる。

【図面の簡単な説明】

【図1】本発明の実施形態に係る情報処理装置の機能構成を示すブロック図である。

【図2】本発明の実施形態の情報処理装置の構成を示すブロック図である。

【図3】本発明の実施形態で実行されるインデックス作成処理を示すフローチャートである。

【図4】本発明の実施形態で実行される登録処理の詳細を示すフローチャートである。

【図5】本発明の実施形態で実行される新キー作成処理の詳細を示すフローチャートである。

【図6】本発明の実施形態で実行される検索処理を示すフローチャートである。

【図7】本発明の実施形態で実行される位置リスト獲得処理の詳細を示すフローチャートである。

【図8】従来の情報処理装置におけるインデックスの概念を示す図である。

【図9】従来の情報処理装置におけるインデックスの概念を示す図である。

【図10】従来の情報処理装置におけるインデックスの概念を示す図である。

【図11】本発明の実施形態のインデックス保持部の詳細な構成を示す図である。

【図12】本発明の実施形態のインデックス保持部の詳細な構成を示す図である。

【図13】本発明の他の実施形態に係る情報処理装置の機能構成を示すブロック図である。

【図14】本発明の他の実施形態に係る情報処理装置の機能構成を示すブロック図である。

【図15】本発明の他の実施形態のインデックスのデータ構造と作成例を示す図である。

【図16】本発明の実施形態を実現するプログラムコー

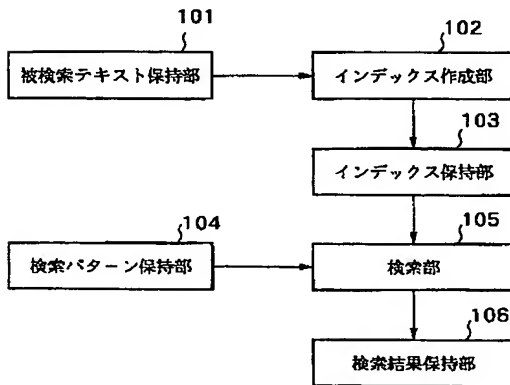
ドを格納した記憶媒体のメモリマップの構造を示す図である。

【図17】本発明の実施形態を実現するプログラムコードを格納した記憶媒体のメモリマップの構造を示す図である。

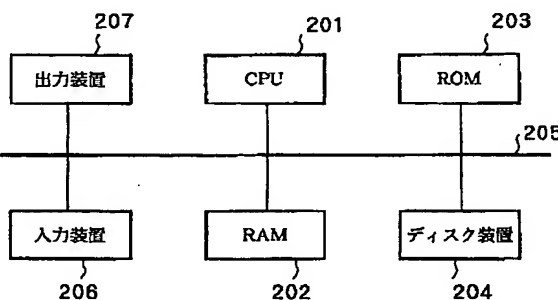
【符号の説明】

- 101 被検索テキスト保持部
- 102 インデックス作成部
- 103 インデックス保持部
- 104 検索パターン保持部
- 105 検索部
- 106 検索結果保持部

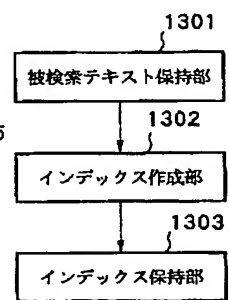
【図1】



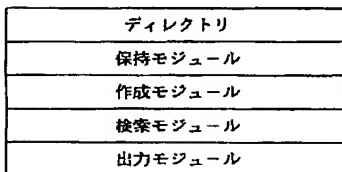
【図2】



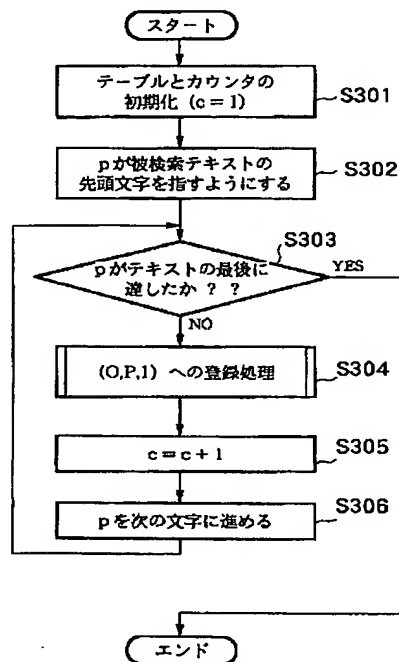
【図13】



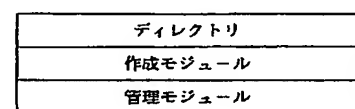
【図16】



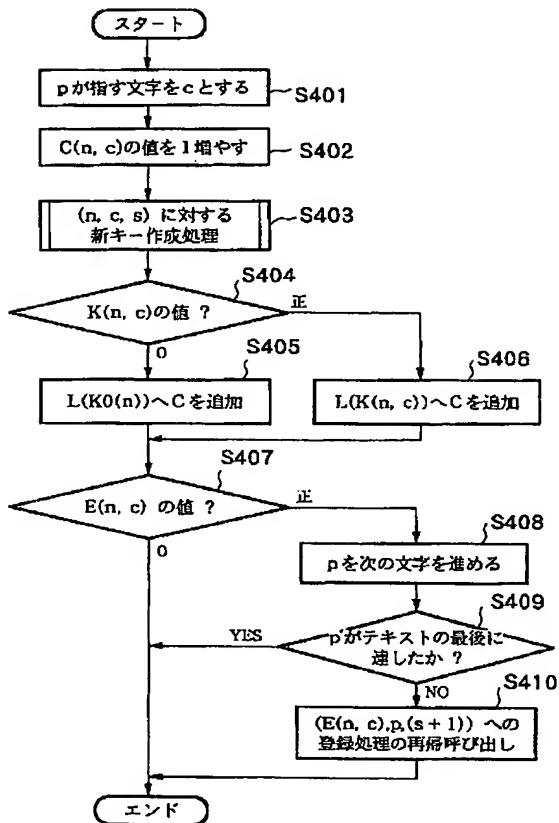
【図3】



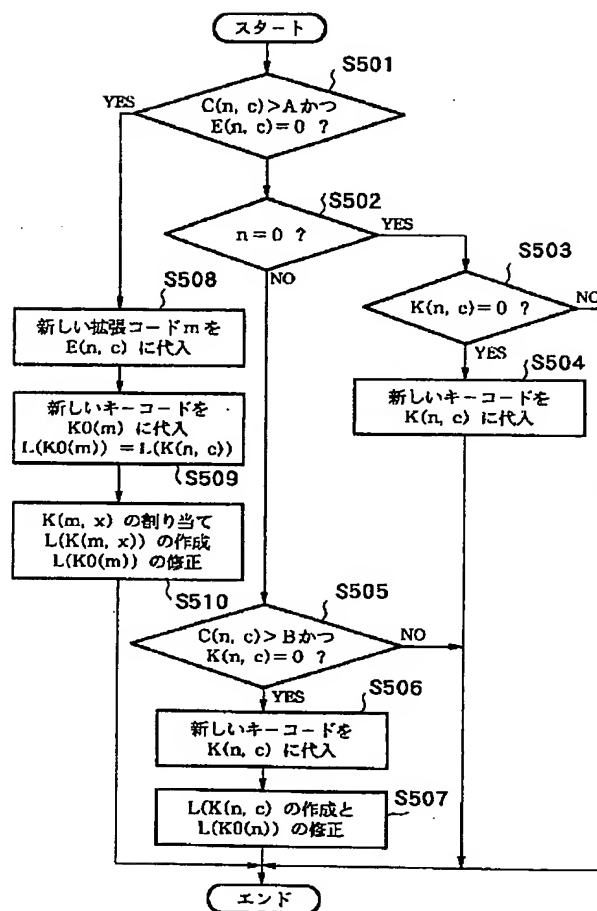
【図17】



【図4】



【図5】



【図8】

の	7
は	14
り	18
れ	20
い	5
ス	4
券	13

キー位置リスト

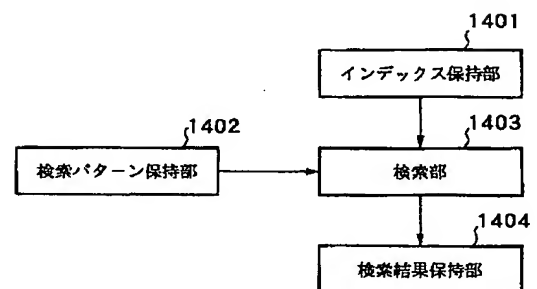
切	19
側	10
対	3
当	11
日	1 8 12 16
売	17
本	2 9 15

キー位置リスト

日券	12
日売	16
日本	1 8
本側	9
本対	2
本日	15

キー位置リスト

【図14】



【図9】

の	7
は	14
り	18
れ	20
い	5
ス	4
券	13

キー位置リスト

切	19
側	10
対	3
当	11
日	1 8 12 16
売	17
本	2 9 15

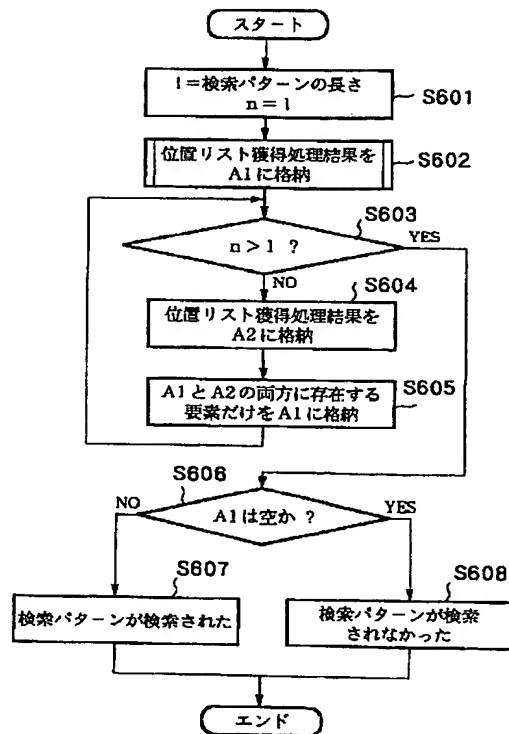
キー位置リスト

日#1	12	16
日#2	1	8
本#1	15	
本#2	2	9

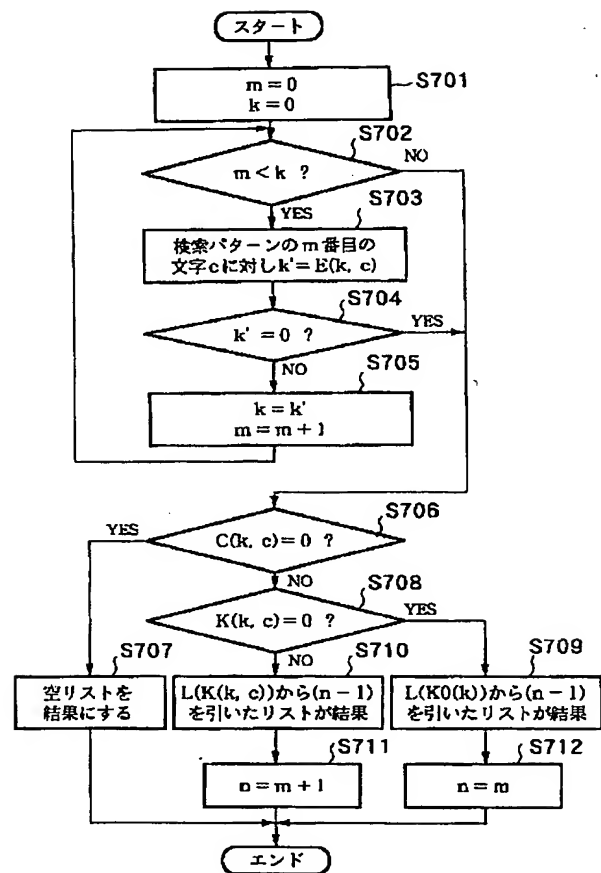
キー位置リスト



【図6】



【図7】



【図10】

の	7
は	14
り	18
れ	20
イ	5
ス	4
券	13

キー位置リスト

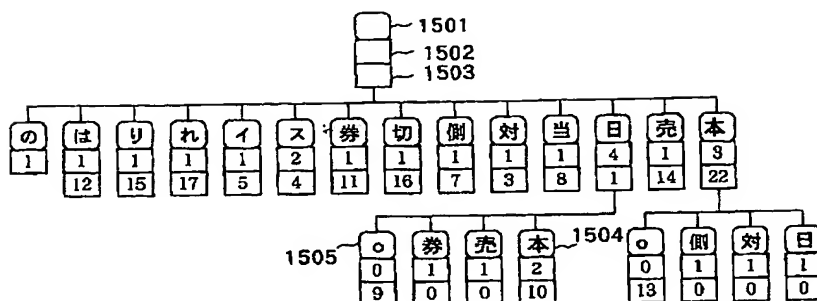
切	19
側	10
対	3
当	11
日	1
売	8
本	12
	16
	17
	2
	9
	15

キー位置リスト

日本	1	8
日(o)	12	16
本(o)	2	9
	15	

キー位置リスト

【図15】



【図11】

出戻回数領域

1101

あ...のはりれイス券切側対当日売本																
0	0	...	1	1	1	1	1	2	1	1	1	1	1	4	1	3
1	0	...	0	0	0	0	0	0	1	0	0	0	0	0	1	2
2	0	...	0	0	0	0	0	0	0	0	1	1	0	1	0	0

キーコード領域 1102

あ...のはりれイス券切側対当日売本																	
0	0	0	...	6	12	15	17	6	4	11	16	7	3	8	1	14	2
1	9	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	10
2	13	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0

拡張コード領域

1103

	あ	...	の	は	り	れ	い	ス	券	切	側	対	当	日	売	本
0	0	...	0	0	0	0	0	0	0	0	0	0	0	1	0	2
1	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置リスト領域

1104

1	1	8	12	16
2	2	9	15	
3	9			
4	4	8		
5	5			
6	7			
7	10			
8	11			
9	12	18		
10	1	8		

11	13			
12	14			
13	2	9	15	
14	17			
15	18			
16	19			
17	20			

【図12】

出現回数領域

1101

あ...のはりれイス券切側対当日売本																
0	0	...	1	0	0	0	1	2	0	0	1	1	1	3	0	2
1	0	...	0	0	0	0	0	0	1	0	0	0	0	0	0	2
2	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0

キーコード領域

1102

あ...のはりれイス券切側対当日売本																	
0	0	0	...	6	0	0	0	5	4	0	0	7	3	8	1	0	2
1	9	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	10
2	0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0

拡張コード領域

1103

あ...のはりれイス券切側対当日売本																
0	0	...	0	0	0	0	0	0	0	0	0	0	0	0	1	0
1	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0
2	0	...	0	0	0	0	0	0	0	0	0	0	0	0	0	0

位置リスト領域 1104

1	1	8	12		
2	2	9			
3	3				
4	4	8			
5	5				
6	7				
7	10				
8	11				
9	12				
10	1	8			

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☒ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**